

# **METHOD TO GENERATE A FORMATTED TRACE FOR AN EMBEDDED DEVICE**

## **Field Of The Invention**

This invention relates to generating a formatted trace for an embedded device.

5

## **Background Of The Invention**

Automated media storage libraries are known for providing cost effective access to large quantities of stored media. These media libraries are often interconnected with one or more host computers. Those one or more host computers write information to one or more information storage media disposed in the library, and/or read information from  
10 those one or more information storage media.

The various components of such media storage libraries often include what are sometimes referred to as “embedded” systems, applications, or devices. Such embedded systems comprise special purposes systems requiring high performance but having relatively few dedicated resources. For example, embedded devices typically comprise  
15 relatively little memory, a low performance processor, few if any standard utilities, and no hard disks.

In addition, embedded devices typically do not comprise a conventional operating system. A conventional operating system is written for flexibility. An embedded system, however, performs a single purpose. Therefore, such an embedded device operates using  
20 a device microcode written to optimize the device’s single function.

A PCI card comprises such an embedded device. Such a PCI card typically includes a processor, a flash memory, and SDRAM. A FICON interface card comprises another such embedded device.

In order to monitor the performance of an embedded device, the device microcode  
5 includes instructions for logging data. In the event of a device error, that logged data can be analyzed to determine the cause of the error. Periodically during the operation of the embedded device, a trace statement is issued which causes, *inter alia*, the parameters the device is executing on at that point in time to be written to a memory device. This trace data is typically written to a revolving trace buffer.

10 If an error is detected, the trace buffer is downloaded, and the trace data is used to determine the source of the error. In order to adequately monitor the performance of such an embedded device, the device microcode must include many trace statements.

What is needed is a method to facilitate the creation of such trace statements, where the output of that trace statement is maximally descriptive. Applicants' method  
15 autogenerates a trace identifier, thereby, simplifying the formation of the corresponding trace statement.

### **Summary Of The Invention**

Applicants' invention includes an apparatus and method to generate a formatted trace for a second device embedded in a first device. The method provides source code  
20 comprising a trace entry, and compiles that source code by the first device to form an embedded device code image comprising a trace description string and a trace description

string address. The method creates a database comprising the trace description string and the trace description string address.

The method uploads to the second device the embedded device code image, and generates trace data using that code. The method then downloads that trace data to the  
5 first device, merges that trace data with the database, and forms a formatted trace.

### **Brief Description Of The Drawings**

The invention will be better understood from a reading of the following detailed description taken in conjunction with the drawings in which like reference designators are used to designate like elements, and in which:

10 FIG. 1 is a flow chart summarizing the steps of Applicants' method;

FIG. 2 shows a trace entry disposed in a source code;

FIG. 3 shows certain trace entry directives;

FIG. 4 shows a trace statement formed using the trace entry of FIG. 2 and the trace entry directives of FIG. 3;

15 FIG. 5 is a block diagram showing an embedded device code image which includes a trace description string and a trace description string address;

FIG. 6 is a block diagram showing a stripped code formed by removing the trace description string from the device code image of FIG. 5, where that trace description string is saved in a database;

20 FIG. 7 is a block diagram depicting uploading the stripped code of FIG. 6 from a control unit to an embedded device;

FIG. 8 is a block diagram depicting downloading trace data from the embedded device to the control unit;

FIG. 9 is a block diagram showing the merger of the trace data of FIG. 8 and the database of FIG. 6;

5        FIG. 10 shows a formatted trace for the embedded device.

### **Detailed Description Of The Preferred Embodiments**

This invention is described in preferred embodiments in the following description with reference to the Figures, in which like numbers represent the same or similar elements.

10        Referring now to FIG. 1, in step 110 Applicants' method provides a source code, where that source code comprises a trace entry. FIG. 2 recites trace entry 200. Trace entry 200 includes trace macro 210, i.e. "TRACE3". Trace entry 200 further includes component ID 220, i.e. OS\_COMP\_ID". Trace entry 200 further includes data 230 which recites the trace entry location, file name, function name, and line number, i.e. "OS\_TRC\_LOCAL\_INFO." Trace entry 200 further includes a function name 240, i.e. "osProcessException." Trace entry 200 further includes a trace description string 250, i.e. "PPC750 exception data; MSR=@w, DSISR=@w, DAR=@w." Trace entry 200 further includes variables 260, 262, and 264.

First trace description string 250 further includes data placeholders 252, 254, and  
20    256. The designator "@w" indicates 32 bits of data. In certain embodiments, one or more of dataplaceholders 252, 254, and/or 256, include the designator "@h," where that designator indicates 16 bits of data. In certain embodiments, one or more of

dataplaceholders 252, 254, and/or 256, include the designator “@b,” where that designator indicates 8 bits of data.

Referring again to FIG. 1, in step 115 Applicants’ method provides trace entry directives. FIG. 3 recites trace entry directives 300. The trace entry directives of step  
5 115 are used by a preprocessor and a compiler disposed in the first device, i.e. in the control unit.

In step 120, Applicants’ method detects the trace entry. In certain embodiments, a pre-processor disposed in the control unit detects the trace entry. In certain embodiments, step 120 further includes detecting the trace macro component of the trace  
10 entry.

In step 125, Applicants’ method compiles the trace entry of step 110 using the trace entry directives of step 115. FIG. 4 illustrates the result of passing the trace statement of FIG. 2 through a pre-processor using the directives of FIG. 3.

In step 130, Applicants’ method takes the address of the trace description string  
15 and assigns that address as the value of the tradeld. In step 135, Applicants’ method forms a device code image for the embedded device. In the illustrated embodiment of FIG. 5, device code image 500 includes instructions 510, 520, 530, 540, and 550. Instructions 510, 520, 530, 540, and 550, are written to instruction memory 501. Instruction 520 references the address for the trace description string. That address is  
20 disposed in a data memory 502, and the memory location starting with that address contains the complete trace description string 410.

In step 140, Applicants' method forms a database which includes both the trace description string and the address for that trace description string. Step 140 further includes saving that database in the first device, i.e. in the control unit.

In step 142, Applicants' method determines whether to form a stripped code by  
5 removing the trace description string from the device code image of step 135. If Applicants' method determines not to form such a stripped code, then the method transitions from step 142 to step 150 wherein the method provides the code image of step 135 to the embedded device. Referring to FIG. 7, in step 150 Applicants' method provides code 500 (FIG. 5) from control unit 710 to embedded device 720. In step 155,  
10 Applicants' method generates trace data using the code image.

If Applicants' method determines in step 142 to form a stripped code, then the method transitions from step 142 to step 145 wherein the method forms a stripped code by removing the trace description string from the device code image of step 135. Referring to FIG. 6, stripped code 600 (FIG. 6) is formed by removing code portion 602,  
15 i.e. address 560 comprising trace description string 410, from device code image 500 (FIG. 5).

In step 152, Applicants' method provides the stripped code of step 145 to the embedded device. Referring to FIG. 7, in step 152 Applicants' method provides code 600 from control unit 710 to embedded device 720. Trace description string 602 (FIG. 6)  
20 is saved in memory 730 disposed in control unit 710.

In step 157, the embedded device generates trace data using the stripped code provided in step 152. Applicants' method transitions from step 157 to step 160 wherein

the method saves the trace data of step 155 / 157 in trace buffer 830 (FIG. 8). In step 165, Applicants' method detects an error in the embedded device. In step 170, Applicants' method provides the data saved in the trace buffer to the control unit. Referring now to FIG. 8, in step 170 the method provides the information saved in trace 5 buffer 830 to control unit 710. Trace buffer portion 832 includes trace entry 840 which comprises address 560.

In step 180, Applicants' method merges the trace data provided in step 170 with the database of step 140. Referring to FIG. 9, in step 180 Applicants' method merges trace entry 840 with trace description string 410 to generate in step 190 formatted trace 10 1000 (FIG. 10).

In certain embodiments, individual steps recited in FIG. 1 may be combined, eliminated, or reordered.

Applicants' invention further includes an article of manufacture comprising a computer useable medium, such as computer useable medium 740 (FIG. 7), having 15 computer readable program code disposed therein method to generate a formatted trace for a second device embedded in a first device using the steps of FIG. 1.

Applicants' invention further includes a computer program product, such as computer program product 750 (FIG. 7), usable with a programmable computer processor having computer readable program code embodied therein method to generate a 20 formatted trace for a second device embedded in a first device using the steps of FIG. 1.

While the preferred embodiments of the present invention have been illustrated in detail, it should be apparent that modifications and adaptations to those embodiments

may occur to one skilled in the art without departing from the scope of the present invention as set forth in the following claims.